# Diagnosis of Discrete-Event Systems by Separation of Concerns, Knowledge Compilation, and Reuse

**Gianfranco Lamperti** and **Marina Zanella** [1]

**Abstract.** Model-based diagnosis of discrete-event systems (DESs) requires the reconstruction of the behavior of the system to be diagnosed, which is computationally expensive and, therefore, time-consuming. Accordingly, most approaches propose a trade-off between off-line and on-line computation: suitable knowledge, derived off-line from the model of the system, can be exploited on-line based on the actual observation. This way, a large amount of model-based reasoning is anticipated off-line, thereby making the on-line task considerably lighter. The essential novelty of this paper, which aims to support the diagnosis of asynchronous DESs, lies in the ability to exploit not only the general-purpose diagnostic knowledge compiled off-line but also the special-purpose knowledge generated on-line for the solution of previous problems, thereby pursuing processing reuse. To this end, compatibility checking is required: the solution of a new diagnostic problem can exploit the solution of another problem provided the latter subsumes the former.

## 1 INTRODUCTION

Discrete-event systems (DESs) [4] are dynamic systems with discrete inputs and outputs, whose behavior can be described in terms of discrete state changes. Since reasoning about discrete models is easier than about continuous ones, from the middle '90s the task of diagnosis of DESs has been receiving an increasing interest [9, 20, 13, 16, 5]. Diagnosing a system means computing its *candidate diagnoses*, each of which is a set of faults that explains the observation collected during the system operation. According to the current shared prospect about diagnosis of DESs, in the general case, the specific faults cannot be inferred without first finding out what has happened to the system over the time interval inherent to the observation [14]. This way, the system evolutions complying with the observation are in fact an output of the diagnostic process, from which candidate diagnoses can be distilled. In this respect, in spite of different terminologies, such as histories [1], situation histories or narratives [3], paths [5], and trajectories [6], all the distinct approaches describe the evolution of a DES as a sequence of state transitions, since the favorite behavioral models of DESs are automata. However, determining the system evolutions is a computationally expensive and, consequently, time-consuming process (see, for instance, [17] about the computational difficulties of the diagnoser approach [18, 19], or the worst case computational complexity analysis in [1], or the discussion in [8]). This is the reason why most of the approaches exploit a trade-off between *off-line* and *on-line* computation, where the former is performed when no diagnostic problem is considered and, therefore, processing is not under the pressure of time constraints, while the latter is performed for solving each specific diagnostic problem given its observation. The rationale is that some kind of knowledge, implicit in the models of the structure and behavior of the system, is compiled off-line in order to speed up on-line processing. This paper applies *knowledge compilation* and *subsumption-based reusability* techniques to the active system approach [1, 2], which deals with diagnosis of a class of DESs, called *active systems* [12], modeled as networks of nondeterministic automata communicating through asynchronous links. The considered task is *a-posteriori* diagnosis: the observation taken as input by the task encompasses the whole *reaction* of an active system, this being a system evolution that is triggered by an external event. Previous approaches to diagnosis of DESs in the literature suggest which (compiled) knowledge to generate off-line by processing the system models and how to exploit it on-line [18, 19, 10, 15, 7]. According to such approaches, compiled knowledge is produced once and for all before any diagnostic problem is considered, in order to set up the diagnostic engine, then such a knowledge is exploited several times on-line, and it never changes. This paper goes beyond the above perspective, by suggesting how to extend the compiled knowledge generated beforehand by progressively adding the knowledge produced on-line for solving diagnostic problems. The proposed shift of perspective triggers further responsibility for the reasoning process, namely ($i$) reusing available knowledge, and ($ii$) generating new knowledge. Diagnostic problems refer to the general notion of an *uncertain observation* [11], which inevitably translates to the expansion of the search space, thereby making reusability an essential requirement for on-line processing.

## 2 SYSTEM MODELING

The compositional model of a *system* is a network of *components* that are connected to one another through *links*. Each component is completely modeled by a *communicating automaton* $\mathcal{C}$ that reacts to events either coming from the external world or from neighboring components through links. Formally, the automaton is a 6-tuple, $\mathcal{C} = (\mathbf{S}, \mathbf{E}_{\rm in}, \mathbf{I}, \mathbf{E}_{\rm out}, \mathbf{O}, \mathbf{T})$, where $\mathbf{S}$ is the set of *states*, $\mathbf{E}_{\rm in}$ the set of *input events*, $\mathbf{I}$ the set of *input terminals*, $\mathbf{E}_{\rm out}$ the set of *output events*, $\mathbf{O}$ the set of *output terminals*, and $\mathbf{T}$ the nondeterministic *transition function*, $\mathbf{T} : \mathbf{S} \times \mathbf{E}_{\rm in} \times \mathbf{I} \times 2^{\mathbf{E}_{\rm out} \times \mathbf{O}} \mapsto 2^{\mathbf{S}}$. A transition $T \in \mathbf{T}$, from state $S$ to state $S'$, which is triggered by event $e$ at input terminal $I$, and generates events $e_1, \ldots, e_k$ at output terminals $O_1, \ldots, O_k$, respectively, is denoted by $T = S \xrightarrow[(e_1,O_1),\ldots,(e_k,O_k)]{(e,I)} S'$. Links, which are the means to store the events exchanged between components, are modeled by a triple $\mathcal{L} = (I, O, M)$, where $I$ is the *input terminal*, $O$ the *output terminal*, and $M$ the *event management*. The latter establishes the internal structure of the link and the effect of each insertion/deletion operation performed on the events temporally saved within it. A specific allocation of events within $L$ is called a *configuration* of $L$. Formally, a system $\Sigma$ is a pair $\Sigma = (\mathbf{C}, \mathbf{L})$, where $\mathbf{C}$ is the set of components and $\mathbf{L}$ the set of links.

**Example 1.** Displayed on top of Fig. 1 are the models *Breaker* and *Protection*. Each model is depicted by the automaton (right) and the set of terminals (left), where input and output terminals are represented as triangles and bullets, respectively. The automaton relevant to the breaker incorporates two states, marked by 0 (closed) and 1 (open), and two transitions, $T_1$ and $T_2$, represented as arrows. When the breaker is closed, either transition $T_1$ or $T_2$ is nondeterministically triggered by event $z$ on input terminal $I$. $T_1$ moves the breaker to state 1 without generating any output event. $T_2$, instead, keeps the state of the breaker unchanged, whilst generating event $f$ at output terminal $O$. The model of the protection embodies four input

[1] Dipartimento di Elettronica per l'Automazione, Università di Brescia, Via Branze 38, 25123 Brescia, Italy. Tel: +390303715596, Fax: +39030380014, Email: lamperti@ing.unibs.it, zanella@ing.unibs.it

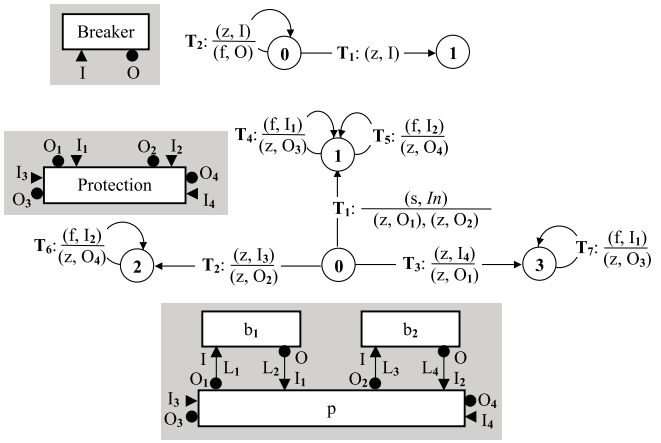


**Figure 1.** Models of breaker and protection (top), and system $\xi$ (bottom).

terminals, $I_1 \cdots I_4$, and four output terminals, $O_1 \cdots O_4$. Terminals $O_1$ and $I_1$ are meant for connection with a breaker on the left, while terminals $O_2$ and $I_2$ are for a communication with a breaker on the right. Instead $I_3$ and $O_3$ allow the protection to exchange events with a neighboring protection on the left. The same applies for $I_4$ and $O_4$, which are a means to communicate with a protection on the right. Depicted on the bottom of Fig. 1 is the topology of a system $\xi$, which integrates protection $p$ and breakers $b_1$ and $b_2$. The protection is connected with the breakers by means of links $L_1 \cdots L_4$.

## 3 BEHAVIOR SPACE

The behavior of a system $\Psi$ may evolve only within a confined space. A *system state* is a pair $\sigma = (\mathbb{S}, \mathbb{L})$, where $\mathbb{S}$ is a record of the states of the components in $\Psi$, while $\mathbb{L}$ is a record of the configurations of the links in $\Psi$. Initially, a system $\Psi$ is in a *quiescent* state $\Psi_0$, wherein all links are empty. Upon the arrival of an event from the external world, $\Psi$ becomes *reacting*, thereby making a series of system transitions, namely a *history* of $\Psi$. Due to asynchronism, each system transition is the transition of one component in $\Psi$. The whole set of possible evolutions of $\Psi$ from the initial state $\Psi_0$ are specified by a graph $Bhv(\Psi, \Psi_0)$, called *behavior space*, whose *extension*, $\|Bhv(\Psi, \Psi_0)\|$, is the whole set of paths from the initial state to a final state, that is, the set of histories of $\Psi$ rooted in $\Psi_0$.

**Example 2.** Shown in Fig. 2 is the behavior space relevant to system $\xi$ (Fig. 1), where the initial state of all components is 0. In each node, the record $\mathbb{S}$ of the component states for $b_1$, $p$, and $b_2$ is on the top, while the record $\mathbb{L}$ of configurations of links $L_1 \cdots L_4$ is on the bottom (incidentally, such configurations involve at most one event per link). Quiescent nodes (double circled) are characterized by the emptiness of the links.

## 4 DIAGNOSTIC PROBLEM

A diagnostic problem $\wp$ for a system $\Psi$ is a 5-tuple, $\wp(\Psi) = (\Psi_0, \mathcal{V}, \mathcal{O}, \mathcal{R}, \mathcal{K})$, where $\Psi_0$ is the *initial state* of $\Psi$, that is, the state of $\Psi$ when the reaction started, $\mathcal{V}$ is the *viewer* (observer) of $\Psi$, with specific visibility properties, $\mathcal{O}$ is the *observation* of the system generated during the reaction, $\mathcal{R}$ is the *ruler*, which establishes the state transitions that are to be considered faulty, and $\mathcal{K}$ is the *knowledge* about the system, including at least the compositional model of $\Psi$ and, possibly, additional *compiled knowledge*. Depending on $\mathcal{K}$, we distinguish two classes of diagnostic problems:

- *Crude problems*, when $\mathcal{K}$ is restricted to the compositional model of $\Psi$, that is, when no compiled knowledge is available;

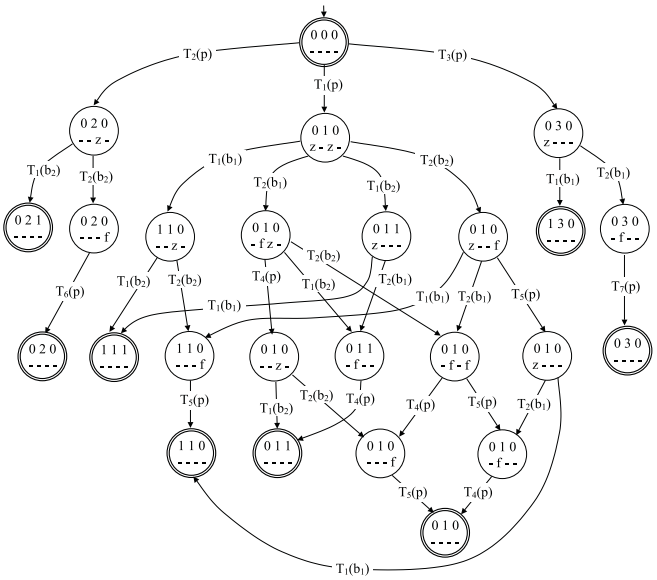


**Figure 2.** Behavior space $Bhv(\xi, \xi_0)$.

- *Compiled problems*, when, besides the model of $\Psi$, $\mathcal{K}$ incorporates compiled knowledge about $\Psi$.

Solving $\wp(\Psi)$ amounts to determining the set of candidate diagnoses, where each diagnosis is the set of faults associated by $\mathcal{R}$ with the transitions in a history $h$ pertinent to $\wp(\Psi)$. The solution of a compiled problem is virtually more efficient than that of a crude problem, since part of the model-based reasoning necessary for solving the problem is somewhat codified in $\mathcal{K}$.

A viewer establishes what component transitions are visible, as well as the specific label generated by each of them. Let $\mathbf{T}$ be the set of transitions relevant to components in $\Psi$, and $\mathbf{V}$ a set of labels including the *null* label $\varepsilon$. A viewer $\mathcal{V}$ is a mapping from $\mathbf{T}$ to $\mathbf{V}$. If $(T, \varepsilon) \in \mathcal{V}$, then $T$ is a *silent* transition, otherwise $T$ is *visible*.

**Example 3.** Considering system $\xi$, a possible viewer $\mathcal{V}_\xi$ for a diagnostic problem $\wp(\xi)$, relevant to the set of labels $\{o_1, o_2, l, r\}$, can be implicitly defined by the set of visible transitions, namely $\mathcal{V}_\xi = \{(T_1(b_1), o_1), (T_1(b_2), o_2), (T_2(p), l), (T_3(p), r)\}$.

An observation $\mathcal{O}$ relevant to a reaction of $\Psi$ is a set of observable events and of their reciprocal temporal order, as perceived by the viewer, typically under uncertainty conditions. As detailed in [11], any $\mathcal{O}$ that belongs to a diagnostic problem is represented by a DAG, the *observation graph*. Each node of the graph corresponds to an observed label, which, however, is *logically uncertain*, that is, it ranges over a set of candidate observable labels (possibly including the null label $\varepsilon$), since the viewer cannot discriminate among them which was the one actually emitted by the system. Edges of the graph specify partial temporal ordering between nodes (*temporal uncertainty*) since, in general, the viewer cannot ascertain the total emission order of the observed labels from their reception order. As such, an observation graph is the formal representation of what has been seen, rather than the result of an elaboration. In order to check the consistency of the system behavior against $\mathcal{O}$, the diagnostic process draws from the given observation graph an additional DAG, namely the *index space* of $\mathcal{O}$, denoted $\mathfrak{I}(\mathcal{O})$, where each path, from the root to a final node, is a *plain observation*, this representing a mode in which labels can be picked up from the nodes of the observation graph based on the partial temporal ordering defined by the edges. Therefore, each plain observation may be the actual sequence of observable labels emitted by the system. The index space includes all and only the plain observations implicitly embedded within an observation graph. A history $h$ of $\Psi$ is said to *comply* with $\wp(\Psi)$ iff,

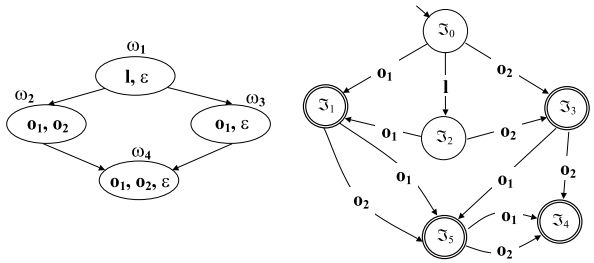


**Figure 3.** Observation $\mathcal{O}_\xi$ for $\xi$ (left) and relevant index space (right).

according to $\mathcal{V}$, the list of observable labels relevant to the transitions in $h$ is a plain observation in $\mathcal{O}$.

**Example 4.** The graph of an observation $\mathcal{O}_\xi$ for $\xi$, as perceived by viewer $\mathcal{V}_\xi$ (Example 3), is depicted on the left of Fig. 3. Next to it is the relevant index space, involving 24 plain observations, for instance, $\langle o_1, o_2\rangle$, which can be derived from the observation graph by picking up labels $\varepsilon$ , $o_1$, $\varepsilon$, and $o_2$ from nodes $\omega_1$, $\omega_2$, $\omega_3$, and $\omega_4$, respectively[2]. According to Fig. 2, the history $\langle T_1(p), T_2(b_1), T_1(b_2), T_4(p)\rangle$ complies with any problem $\wp(\xi) = (\xi_0, \mathcal{V}_\xi, \mathcal{O}_\xi, \mathcal{R}, \mathcal{K})$, where $\xi_0$ involves all components in state 0.

A ruler establishes what transitions are faulty. Let $\mathbf{T}$ be the set of transitions in $\Psi$, and $\mathbf{R}$ a set of labels including the *null* label $\varepsilon$. A ruler $\mathcal{R}$ is a mapping from $\mathbf{T}$ to $\mathbf{R}$. If $(T, \varepsilon) \in \mathcal{R}$, then $T$ is *normal*, otherwise $T$ is *faulty*. A history $h$ of $\Psi$ is said to *imply* a diagnosis $\delta$, where $\delta$ is the set of faults associated with the faulty transitions of $h$ defined in $\mathcal{R}$.

**Example 5.** $\mathcal{R}_\xi = \{(T_1(p), s), (T_2(b_1), f_1), (T_2(b_2), f_2)\}$ is a ruler for $\xi$. Accordingly, considering Fig. 2, the history $\langle T_1(p), T_2(b_1), T_1(b_2), T_4(p)\rangle$ implies the diagnosis $\{s, f_1\}$.

Three classes of compiled-knowledge are envisaged, which are instantiated by relevant *knowledge graphs*, namely:

- *Behavior*, a graph whose extension[3] is a subset of the extension of the behavior space;
- *Abduction*, a graph whose extension is a subset of the extension of the behavior space and where each final node is decorated with the diagnosis implied by each relevant history, in accordance with a ruler $\mathcal{R}$;
- *Map*, a graph where each path is the list of observable events relevant to one or several histories of $\Psi$, in accordance with a viewer $\mathcal{V}$, and each final node is marked by a set of diagnoses, according to a ruler $\mathcal{R}$.

From a formal viewpoint, the solution of a diagnostic problem $\wp(\Psi)$ depends on $\Psi_0$, $\mathcal{V}$, $\mathcal{O}$, and $\mathcal{R}$ only, as the role of $\mathcal{K}$ is to speed up the diagnostic process, without any influence on the result. According to our terminology, the solution of $\wp(\Psi)$, denoted $\Delta(\wp(\Psi))$, is the set of diagnoses that are implied by the histories that comply with $\wp(\Psi)$.

## 5 SOLVING CRUDE PROBLEMS

The solution of a crude problem $\wp(\Psi)$ is performed on-line by first generating the abduction $Abd(\wp(\Psi))$ and, then, by collecting the diagnoses marking the final nodes.

**Example 6.** Consider a problem $\wp(\xi) = (\xi_0, \mathcal{V}_\xi, \mathcal{O}_\xi, \mathcal{R}_\xi, \mathcal{K})$, where $\mathcal{V}_\xi$, $\mathcal{O}_\xi$, and $\mathcal{R}_\xi$ are defined in Examples 3, 4, and 5, respectively. The abduction $Abd(\wp(\xi))$ is displayed in Fig. 4, where each final node is marked by the corresponding diagnosis (set of faults based on $\mathcal{R}_\xi$). Thus, $\Delta(\wp(\xi)) = \{\emptyset, \{s\}, \{s, f_1\}, \{s, f_2\}\}$.

[2] A technique for generating the index space is described in [11].

[3] The notion of extension introduced in Section 3 can be applied to DAGs.

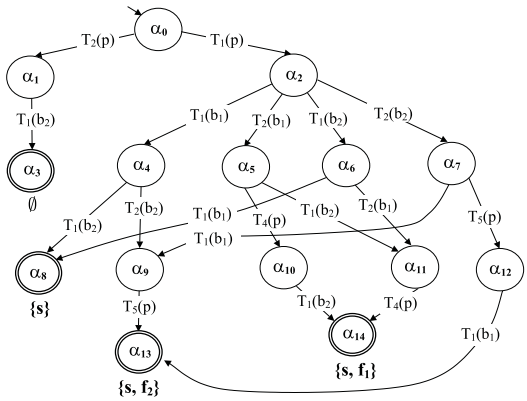


**Figure 4.** Abduction $Abd(\wp(\xi))$.

## 6 PREPROCESSING

The sort of compiled knowledge $\mathcal{K}$ relevant to $\wp(\Psi)$ is twofold:

- *General-purpose knowledge* compiled off-line independently of any specific observation;
- *Special-purpose knowledge* generated for solving specific diagnostic problems.

Preprocessing concerns the generation of the former. Each general-purpose graph can be either a *behavior space*, an *abduction space*, or a *map space*, namely, $Bhv(\psi, \psi_0)$, $Abd(\psi, \psi_0, \mathcal{R})$, $Map(\psi, \psi_0, \mathcal{V}, \mathcal{R})$, respectively, where $\psi$ is a subsystem of $\Psi$ ($\Psi$ is recursively broken down into subsystems and knowledge is possibly compiled for them).

**Example 7.** Shown in Fig. 5 is the map space $Map(\xi, \xi_0, \mathcal{V}_\xi, \mathcal{R}_\xi)$. Incidentally, all nodes are final. Thus, based on $\mathcal{V}_\xi$, each path rooted in $\mu_0$ is a possible sequence of observable labels generated by (possibly several) histories in $Bhv(\xi, \xi_0)$ (see Fig. 2).

## 7 SOLVING COMPILED PROBLEMS

Solving a compiled diagnostic problem amounts to a pattern-matching between the index space of the observation and a suitable knowledge graph, whether general-purpose or special-purpose. Depending on whether the available graph is a map, an abduction, or a behavior, the problem is either classified as a *$\mu$-problem*, an *$\alpha$-problem*, or a *$\beta$-problem*, respectively. Generally speaking, $\mu$-problems can be solved more efficiently than $\alpha$-problems, which in turn can be solved more efficiently than $\beta$-problems. The essential point is to check the compatibility between the problem $\hat{\wp}(\psi) = (\psi_0, \hat{\mathcal{V}}, \hat{\mathcal{O}}, \hat{\mathcal{R}}, \hat{\mathcal{K}})$ to be solved and a knowledge graph. A knowledge graph can be generically denoted as a function $\gamma(\psi, \psi_0, \mathcal{V}, \mathcal{O}, \mathcal{R})$, where each of $\mathcal{V}$, $\mathcal{O}$, and $\mathcal{R}$ is possibly null. Specifically, if the graph is general-purpose, the observation $\mathcal{O}$ will be null, meaning that $\gamma$ is not constrained by any observation. If $\gamma$ is special-purpose, it will refer to a previously-solved problem $\wp(\psi)$, and, therefore, to a specific observation. If $\gamma$ is an abduction space, $\mathcal{V}$ will be null. If it is a behavior space, $\mathcal{R}$ will be null too.

Compatibility checking is based on *subsumption relationships* between viewers, observations, and rulers. Intuitively, if such relationships are met, $\gamma$ will contain all the patterns necessary to solve $\hat{\wp}(\psi)$. For instance, if $\gamma$ is a map, $\|\gamma\|$ (the set of paths in $\gamma$) will be a superset of the sequences of observable labels generated by the histories of the behavior relevant to $\hat{\wp}(\psi)$. As such, $\gamma$ can be exploited for yielding a sound and complete solution of $\hat{\wp}(\psi)$, without any (on-line) reconstruction of the system behavior.

Let $\mathcal{O}$ and $\mathcal{O}'$ be two observations for $\psi$. We say that $\mathcal{O}$ *subsumes* $\mathcal{O}'$, denoted $\mathcal{O} \Supset \mathcal{O}'$, iff $\|\mathfrak{I}(\mathcal{O})\| \supseteq \|\mathfrak{I}(\mathcal{O}')\|$.

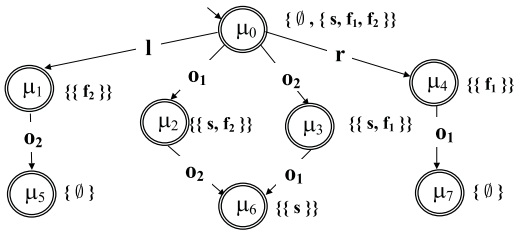


**Figure 5.** Map space $Map(\xi, \xi_0, \mathcal{V}_\xi, \mathcal{R}_\xi)$.

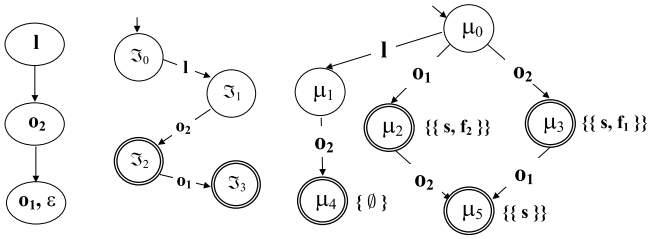


**Figure 6.** Observation $\mathcal{O}'_\xi$ for $\xi$, $\mathfrak{I}(\mathcal{O}'_\xi)$, and $Map(\wp(\xi))$.

**Example 8.** Shown in Fig. 6 is the graph of an observation $\mathcal{O}'_\xi$ for system $\xi$, along with its index space. Clearly, $\mathcal{O}_\xi \Supset \mathcal{O}'_\xi$, since

$$\|\mathfrak{I}(\mathcal{O}_\xi)\| \supset \left(\|\mathfrak{I}(\mathcal{O}'_\xi)\| = \{\langle l, o_2\rangle, \langle l, o_2, o_1\rangle\}\right).$$

Subsumption for viewers is defined as follows. Let $\mathcal{V}$ and $\mathcal{V}'$ be two viewers for $\psi$, and $\mathbb{T}$ and $\mathbb{T}'$ the relevant sets of visible transitions, respectively. Then, $\mathcal{V} \Supset \mathcal{V}'$ iff $(i)$ $\mathbb{T} \supseteq \mathbb{T}'$, and $(ii)$ for each label $\ell$ relevant to a transition in $\mathbb{T}$ that is also in $\mathbb{T}'$, there exists a label $\ell'$ relevant to $\mathcal{V}'$ such that the set of transitions associated with $\ell$ in $\mathcal{V}$ is contained in the set of transitions associated with $\ell'$ in $\mathcal{V}'$. Subsumption between rulers is defined in the same way.

**Example 9.** Consider $\mathcal{V}_\xi$ (Example 3) and $\mathcal{V}'_\xi = \{(T_1(b_1), open), (T_1(b_2), open), (T_2(p), left)\}$. Accordingly, $\mathcal{V}_\xi \Supset \mathcal{V}'_\xi$.

Subsumption between viewers (rulers) allows the reuse of graphs by means of a simple renaming operation, which is essentially a projection of the labels of one graph into the labels of the other.

When the knowledge graph $\gamma$ is special-purpose, the compatibility check translates to checking the subsumption between the problem to be solved and the problem relevant to $\gamma$. Both *weak subsumption* and *strong subsumption* are introduced.

Let $\wp = (\mathcal{K}, \psi_0, \mathcal{V}, \mathcal{O}, \mathcal{R})$ and $\wp' = (\mathcal{K}', \psi_0, \mathcal{V}', \mathcal{O}', \mathcal{R}')$ be two problems for $\psi$. We say that $\wp$ *weakly subsumes* $\wp'$, denoted $\wp \sqsupseteq \wp'$, iff $\mathcal{V} \Subset \mathcal{V}'$ and $\mathcal{O} \Supset \mathcal{O}'_{[\mathcal{V}]}$, where $\mathcal{O}'_{[\mathcal{V}]}$ is the projection of $\mathcal{O}'$ on the labels of $\mathcal{V}$. We say that $\wp$ *strongly subsumes* $\wp'$, denoted $\wp \Supset \wp'$, iff $\wp \sqsupseteq \wp'$ and $\mathcal{R} \Supset \mathcal{R}'$.

Such subsumptions are characterized by the *co-variance* of both observation and ruler, and the *contra-variance* of the viewer. The definition of problem subsumption may sound odd, especially because of the contra-variance property of the viewer. Intuitively, it prevents observation $\mathcal{O}$ to be more constrained than $\mathcal{O}'$. Consequently, the behavior relevant to $\wp$ will incorporate all the histories of the behavior relevant to $\wp'$. As such, the behavior of $\wp$ may be reused to solve $\wp'$. Furthermore, if strong subsumption holds, knowledge reusability may be extended to the graphs involving diagnostic information, namely abductions and maps.

**Example 10.** Consider the problem $\wp(\xi)$ defined in Example 6. Assume the new problem $\wp'(\xi) = (\xi_0, \mathcal{V}_\xi, \mathcal{O}'_\xi, \mathcal{R}_\xi, \mathcal{K}')$, where $\mathcal{O}'_\xi$ is defined in Example 8 (Fig. 6). Accordingly, $\wp(\xi) \Supset \wp'(\xi)$.

Diagnosis reuse applies as well when the available knowledge is relevant to a different system that is *isomorphic* to the current one. Two subsystems $\psi$ and $\psi'$ are isomorphic when their topologies match and corresponding component/link models are the same, respectively. This is not unusual in real, complex systems, which are made up by the composition of similar (isomorphic) subsystems. The regularity of the system is bound to support diagnosis reuse.

Once a compatible knowledge-graph $\gamma$ has been selected based on the given subsumption relationships, the actual computation of the candidate diagnoses relevant to the new problem $\hat{\wp}$ can be performed by a *matching* operation between $\gamma$ and the observation $\hat{\mathcal{O}}$. Such a matching has three variants, depending on whether $\gamma$ is a map, an abduction, or a behavior. Specifically, if $\gamma$ is a map, the solution of $\hat{\wp}$ will translate to the operation $\Delta(\hat{\wp}) = Cand(\gamma \asymp \hat{\mathcal{O}})$, where $\asymp$ is the matching operator and $Cand$ collects the candidate diagnoses associated with the final states of the map resulting from the matching. The matching yields a graph where each node is a pair $(\mu, \Im)$, where $\mu$ is a node of the map and $\Im$ a node of $\mathfrak{I}(\mathcal{O})$. The matching is driven by the equality of the labels within the two graphs.

**Example 11.** Consider the map space in Fig. 5, $Map(\xi, \xi_0, \mathcal{V}_\xi, \mathcal{R}_\xi)$, where $\mathcal{V}_\xi$ and $\mathcal{R}_\xi$ are defined in Examples 3 and 5, respectively. Assume the problem $\wp(\xi) = (\xi_0, \mathcal{V}_\xi, \mathcal{O}_\xi, \mathcal{R}_\xi, \mathcal{K})$ solved in Example 6. Since the map space complies with $\xi_0$, $\mathcal{V}_\xi$, and $\mathcal{R}_\xi$, the same problem can be solved by the matching $Map(\xi, \xi_0, \mathcal{V}_\xi, \mathcal{R}_\xi) \asymp \mathcal{O}_\xi$, which involves final states $\mu_2$, $\mu_3$, $\mu_5$, and $\mu_6$ (see Fig. 5), thereby giving rise to the same set of candidate diagnoses as in Example 6, namely

$$\Delta(\wp(\xi)) = \{\emptyset, \{s\}, \{s, f_1\}, \{s, f_2\}\}.$$

**Example 12.** The solution of $\wp(\xi)$ in Example 11 is based on the exploitation of the general-purpose knowledge provided by the map space shown in Fig. 5. Now we consider the diagnostic problem $\wp'(\xi)$ defined in Example 10, assuming that the available knowledge only incorporates the (special-purpose) graphs inherent to the solution of the crude problem $\wp(\xi)$ (behavior, abduction, and map). (Note that, when solving a crude problem, the generation of the relevant map can be deferred off-line based on the available abduction.) Specifically, $Map(\wp(\xi))$ is outlined on the right of Fig. 6. In contrast with $Map(\xi, \xi_0, \mathcal{V}_\xi, \mathcal{R}_\xi)$ in Fig. 5, not all the states are final. Since $\wp(\xi) \Supset \wp'(\xi)$, the solution of $\wp'(\xi)$ can be determined by a simple matching between $Map(\wp(\xi))$ and $\mathcal{O}'_\xi$ (see $\mathfrak{I}(\mathcal{O}'_\xi)$ in Fig. 6), which leads to state $\mu_4$ (marked by the empty diagnosis), and distilling the candidate diagnoses marking such a state, namely:

$$\Delta(\wp'(\xi)) = Cand\left(Map(\wp(\xi)) \asymp \mathcal{O}'_\xi\right) = \{\emptyset\}.$$

## 8 CONCLUSION

This paper proposes both the modeling primitives and the reasoning mechanisms for performing a-posteriori diagnosis of a class of DESs by exploiting knowledge compilation and knowledge reuse. A modeling novelty consists in decoupling the (behavioral) models of system components from the descriptions of their observability and abnormality properties. In the literature, only a more limited separation between component models and observability properties can be found: in [5] each specific problem assigns the same observability to all the instances of the same component type, whereas each instance can be given distinct properties in the current approach. As to the separation between component models and abnormality properties, this is exclusive to the approach described in the present paper.

Separation of concerns, besides opening the way to a systematic study of the diagnosability properties of a given artifact, makes the behavioral models of components completely reusable across several diagnostic problems and contexts, characterized by distinct abilities to detect output events and/or by distinct diagnostic intents. Moreover, although not emphasized in this paper, both off-line and on-line generation of knowledge can be performed in a modular way, as in [2, 11], by decomposing a problem into subproblems, where each subproblem can possibly be solved by reusing available knowledge.

In this scenario, separation of concerns allows for a flexible formulation of subproblems, so as to optimize knowledge reuse.

Reusability applies to both general (observation-independent) and specific (observation-dependent) knowledge. Solving a diagnostic problem inherent to a system typically amounts to processing on-line existing knowledge relevant to the given system or an isomorphic one, provided such a knowledge contains, either directly or indirectly, all the evolutions complying with the given observation. This can be checked by means of defined subsumption conditions.

For the sake of shortness, the paper provides a simplified view of the method. Indeed, the complete method, in order to solve a diagnostic problem inherent to a system, can reuse also available knowledge inherent to one of its supersystems (or to a system which is isomorphic to a supersystem). Besides, the complete approach both extends the concept of isomorphism to diagnostic problems and provides a formal method to detect isomorphic problems, i.e. to recognize in advance, given the specifications of two problems, whether the solution of the former can be mapped to that of the latter via renaming. This ability translates to an increase in efficiency of knowledge compilation and exploitation.

The proposed diagnostic method includes as a particular case the previous approach to diagnosis of active systems with uncertain observations [11], which relies on the availability of component models only. However, the new method surpasses the previous one in several aspects: while performing either on-line or off-line reasoning, it can generate intermediate chunks of knowledge for reuse and produce the abduction inherent to the considered system, which includes candidate diagnoses, without any need for their successive heavy distillation, as instead required in [1, 2, 11]. Besides, in case an abduction has been generated on-line (and the candidate diagnoses have been provided as output based on it), the corresponding map can be drawn off-line, thus generating knowledge that allows for more efficient future reasoning sessions.

No previous contributions in the literature face the topic of on-line generation of compiled knowledge for diagnosis nor that of on-line knowledge reuse based on subsumption rules. In fact, knowledge compilation [18, 19, 10, 15, 7] has so far been confined to an off-line activity only, whose result can be used for solving any diagnostic problem relevant to the considered system. Moreover, the concept of an observation adopted in the current paper is the most expressive in the literature on diagnosis of DESs.

In order to make a more detailed comparison with the diagnoser approach [18, 19], which was the first to address the trade-off between off-line and on-line computation, it is worth recalling that it deals with synchronous DESs while our approach takes into account asynchronous DESs. A further major difference, besides those highlighted above as holding in general, is that the current work avoids generating any global system model as well as any global diagnoser, which instead are both created by the diagnoser approach. Indeed, the map space of the current approach may resemble the diagnoser of the diagnoser approach, however some meaningful points can be singled out to substantiate the former allegation:

- A map space can be generated by following a recursive incremental strategy that reduces the size of the search space and is amenable to a parallel execution while the diagnoser approach does not benefit of any problem-decomposition/solution-composition strategy;
- A map space inherent to a system can be used for solving diagnostic problems inherent to any isomorphic system (as well as to any of its subsystems) while the diagnoser is dedicated to one system;
- A map space, unlike the diagnoser, does not include any ambiguous candidate diagnosis.

In [15] the use of local diagnosers, after the (global) diagnoser approach [18, 19], is combined with the ability to adopt a problem-decomposition/solution-composition paradigm, as in the active system approach [1]. The method amounts to exploiting the knowledge compiled off-line (only), i.e. the diagnosers, to perform the on-line modular reconstruction of the behavior of the considered system. This resembles the on-line exploitation of general purpose knowledge of the current approach. However, while the compiled knowledge of [15] necessarily refers to a single component, the general purpose knowledge of the current approach may be inherent to a whole (sub)system and is threefold in nature. Moreover, ($i$) our approach can exploit on-line not only general knowledge but also specific knowledge, and ($ii$) it can produce off-line not only general knowledge but also specific knowledge.

A challenge for future research on diagnosis of active systems according to the method described in this paper is to find out specific techniques for efficiently checking observation subsumption as well as to provide an automatic support for continuous indexing and maintenance of (graph-based) knowledge.

## REFERENCES


[1] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, 'Diagnosis of large active systems', *Artificial Intelligence*, **110**(1), 135–183, (1999).

[2] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, 'Diagnosis of a class of distributed discrete-event systems', *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* , **30**(6), 731–752, (2000).

[3] C. Barral, S. McIlraith, and T.C. Son, 'Formulating diagnostic problem solving using an action language with narratives and sensing', in *Seventh International Conference on Knowledge Representation and Reasoning – KR'2000*, pp. 311–322, Breckenridge, Colorado, (2000).

[4] C.G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, volume 11 of *The Kluwer International Series in Discrete Event Dynamic Systems*, Kluwer Academic Publisher, Boston, MA, 1999.

[5] L. Console, C. Picardi, and M. Ribaudo, 'Process algebras for systems diagnosis', *Artificial Intelligence*, **142**(1), 19–51, (2002).

[6] M.O. Cordier, A. Grastien, C. Largouët, and Y. Pencolé, 'Efficient trajectories computing exploiting inversibility properties', in *Fourteenth International Workshop on Principles of Diagnosis –DX'03* , pp. 93–98, Washington DC, (2003).

[7] R. Garatti, G. Lamperti, and M. Zanella, 'Diagnosis of discrete-event systems with model-based prospection knowledge', in *ECAI 2002. Proceedings of the 15th European Conference on Artificial Intelligence* , ed., F. van Harmelen, 427–431, IOS Press, Amsterdam, NL, (2002).

[8] J. Kurien and P.P. Nayak, 'Back to the future for consistency-based trajectory tracking', in *Eleventh International Workshop on Principles of Diagnosis – DX'00*, pp. 92–100, Morelia, MX, (2000).

[9] G. Lamperti and P. Pogliano, 'Event-based reasoning for short circuit diagnosis in power transmission networks', in *Fifteenth International Joint Conference on Artificial Intelligence – IJCAI'97* , pp. 446–451, Nagoya, J, (1997).

[10] G. Lamperti and M. Zanella, 'Generation of diagnostic knowledge by discrete-event model compilation', in *Seventh International Conference on Knowledge Representation and Reasoning – KR'2000* , pp. 333–344, Breckenridge, Colorado, (2000).

[11] G. Lamperti and M. Zanella, 'Diagnosis of discrete-event systems from uncertain temporal observations', *Artificial Intelligence*, **137**(1–2), 91–163, (2002).

[12] G. Lamperti and M. Zanella, *Diagnosis of Active Systems – Principles and Techniques*, volume 741 of *The Kluwer International Series in Engineering and Computer Science* , Kluwer Academic Publisher, Dordrecht, NL, 2003.

[13] J. Lunze, 'Diagnosis of quantized systems based on a timed discrete-event model', *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* , **30**(3), 322–335, (2000).

[14] S.A. McIlraith, 'Explanatory diagnosis: conjecturing actions to explain observations', in *Sixth International Conference on Principles of Knowledge Representation and Reasoning – KR'98* , pp. 167–177, Trento, I, (1998).

[15] Y. Pencolé, 'Decentralized diagnoser approach: application to telecommunication networks', in *Eleventh International Workshop on Principles of Diagnosis – DX'00*, pp. 185–192, Morelia, MX, (2000).

[16] Y. Pencolé, M.O. Cordier, and L. Rozé, 'Incremental decentralized diagnosis approach for the supervision of a telecommunication network', in *Twelfth International Workshop on Principles of Diagnosis – DX'01* , pp. 151–158, San Sicario, I, (2001).

[17] L. Rozé, 'Supervision of telecommunication network: a diagnoser approach', in *Eighth International Workshop on Principles of Diagnosis – DX'97*, pp. 103–111, Mont St. Michel, F, (1997).

[18] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C. Teneketzis, 'Diagnosability of discrete-event systems', *IEEE Transactions on Automatic Control*, **40**(9), 1555–1575, (1995).

[19] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C. Teneketzis, 'Failure diagnosis using discrete-event models', *IEEE Transactions on Control Systems Technology* , **4**(2), 105–124, (1996).

[20] S.H. Zad, R.H. Kwong, and W.M. Wonham, 'Fault diagnosis in timed discrete-event systems', in *American Control Conference* , pp. 1756–1761, Pheonix, AZ, (1999).